# From CAD to Adapted Solution for Error Controlled CFD Simulations

**M. Delanaye, A. Patel**
NUMECA International S.A.
Avenue Franklin Roosevelt, 5
B-1050 Brussels, Belgium

e-mail: michel.delanaye@numeca.be
web page: http://www.numeca.com

**K. Kovalev, B. Léonard, Ch. Hirsch**
Vrije Universiteit Brussel
Pleinlaan 2
1050 Brussels
Belgium

The aim of this paper is to present an integrated environment (FINE/Hexa$^{TM}$, HEXPRESS$^{TM}$) for the error controlled simulation of industrial flows in complex geometries. The approach uses hexahedral unstructured meshes to ensure accurate solutions and mesh adaptation to optimize mesh resources. Initial hexahedral coarse meshes are automatically generated for complex domains with minor CAD model manipulation thanks to a volume to surface mesh generation approach. A multigrid method tightly coupled with the mesh adaptation history allows the fast resolution of the non-linear discrete flow problem resulting from a second-order cell-centered approximation.

## INTRODUCTION

There is still an important effort to make for Computational Fluid Dynamics to become routinely used and trusted in the industrial design environment. Industrials are faced with extremely complex flows within complicated geometries. Besides real physics modeling and turbulence flow aspects, which still remain exploratory domains, it is our duty to provide industry with numerical tools capable of accurately solving the Navier-Stokes equations. The goal is to perform accurate CFD simulations for a new geometry in less than 24 hours with reasonably sized computers, with most of the time being spent for the flow computation. This means that, compared to current CFD tools, it is necessary to reduce the time for importing CAD model in grid generators, limit grid generation time to an hour and minimize computational time by optimizing the grid size for a given flow problem. Error controlling also plays an essential role to gain trust. The present approach accounts for this aspect by emphasizing on mesh adaptation to optimize mesh resources and accuracy in the intricate flow regions.

The first aspect of this quest is the interpretation of CAD models in grid generators. CAD definition of a model is usually poorly defined. Encountered problems are related to overlapping NURBS patches or holes and faults in the geometry definition, etc. Most of them can be attributed to the surface modeling paradigm or to the multiple translations between various formats equipped with different tolerances. A Parasolid$^{TM}$ CAD engine is integrated in the NUMECA generator HEXPRESS$^{TM}$ which allows to automatically import solid models generated with this engine. In case the CAD model is unclean, it is necessary to employ a CAD repair system in order to create a water-tight volume. The computational domain supported by HEXPRESS$^{TM}$ is a triangulated representation of the CAD model. Each surface is supported by a triangulation whose unique purpose is to define the geometry. There is no requirement on its quality except that it has to approximate the geometry sufficiently well. The mesh generation procedure is a top-down approach where the volume mesh is directly created without any reference to a surface mesh. This is an advantage compared to other more common unstructured meshing approaches, which usually require a large human investment in the definition of a surface triangulation compatible with the CFD simulation.

In this work, we choose to exploit the potential of hexahedral unstructured meshes (Schneiders, 1996) although they are much less popular than hybrid tetrahedral prismatic grids because of their inherent topological difficulties to mesh complex geometries. Hexahedral meshes potentially offer higher accurate solutions than tetrahedral meshes when using classical numerical methods. It is the best choice for resolving highly sheared flows such as boundary layers. The computational domain is initially covered with a structured mesh corresponding to the bounding box of the domain. This initial mesh, which does not

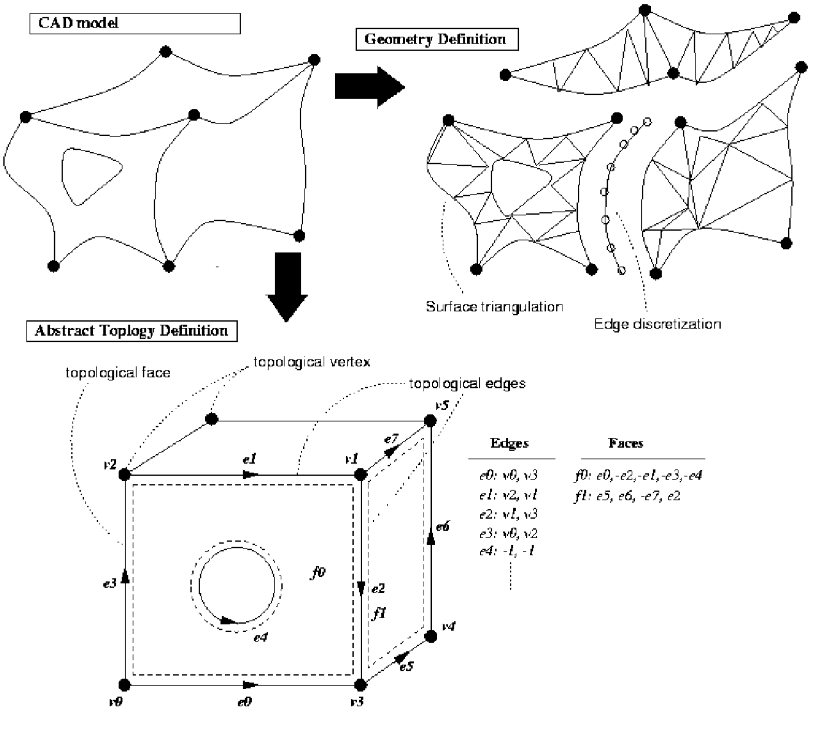
*Paper presented at the RTO AVT Symposium on "Reduction of Military Vehicle Acquisition Time and Cost through Advanced Modelling and Virtual Simulation", held in Paris, France, 22-25 April 2002, and published in RTO-MP-089.*

conform to the geometry, is successively refined anisotropically in order for the cell sizes to match the geometry length scales. Unlike similar Cartesian based methods (Aftosmis, 1997), we do not cut the cells intersecting the geometry in arbitrary polyhedra since they are impossible to use for high-Reynolds number flow simulations. Those cells are removed from the initial grid. Reconnecting the remaining staircase shape of the volume grid to the geometry is a challenging issue. Several procedures have been proposed such as the creation of hybrid grids intersected with the remaining Cartesian volume grid, or overlaid with the latter by a Chimera procedure, or finally connected to it by a tetrahedralization of the remaining gaps. The present technique differs from the latter by directly fitting the non-body fitted Cartesian grid to the domain boundaries using a snapping method (Taghavi, 1996). Sophisticated algorithms are implemented to recover the geometry features such as corners and ridges not preserved by the surface snapping.

The flow solver is tightly connected to the mesh generator by sharing common C++ classes and can therefore benefit from the mesh generator cell subdivision machinery to perform aggressive adaptation of the mesh to the flow solution. It is based on a finite volume cell centered approach. Space discretization is based on the classical Jameson-type centered scheme augmented by blended second and fourth order scalar dissipation. Fast convergence to steady state solutions is obtained thanks to an explicit Runge-Kutta scheme accelerated by a multigrid strategy. This method is combined with a second-order backward time-integration through a dual time-stepping approach for unsteady computations. The Spalart-Allmaras model and several variants of the k-ε model have been implemented to simulate turbulent flows.

# CAD model

The starting point of any simulation is the definition of an appropriate computational domain, which, in most cases, can be interpreted as the complementary of the solid parts present in the model. HEXPRESS$^{TM}$ expects a water-tight computational domain. In fact, it is equipped with a topology and a geometry part as presented in Figure 1. The topology describes the skeleton of the model. Basically it allows to define a closed volume. It thus provides information on the connection of the model surfaces (topological faces) trough common curves (topological edges). Similarly, it also connects curves together by common corners (topological vertices). The geometry part defines the actual geometry of the model. Each surface of the model is described by a triangulation; each curve by a list of points connected by segments and the corners eventually are defined by a single point.



| Edges | Faces |
|---|---|
| e0: v0, v3 | f0: e0,-e2,-e1,-e3,-e4 |
| e1: v2, v1 | f1: e5, e6, -e7, e2 |
| e2: v1, v3 | |
| e3: v0, v2 | |
| e4: -1, -1 | |

**Figure 1: HEXPRESS$^{TM}$ computational domain definition.**

HEXPRESS™ is equipped with a Parasolid™ CAD engine. Thus, any CAD model native from this engine is transparently loaded by the mesh generator. Other native model must be translated to the Parasolid™ format before being processed.

A CAD model usually exhibits complex features which are useless for the flow simulation. It is therefore wise to remove these geometry details from the model before proceeding to mesh generation. In HEXPRESS™, such removal is not applied to the model directly but the complexity of the computational domain is simplified by merging some surfaces. The edges in common between the merged surfaces are then removed and are not captured in the mesh. This merge is performed at the topology level, hence no NURBS or surface representation is reconstructed. The solution of HEXPRESS™ to the geometry simplification is therefore simple and fast. Figure 2 presents the computational domain of a draft tube before (left) and after (right) simplification.

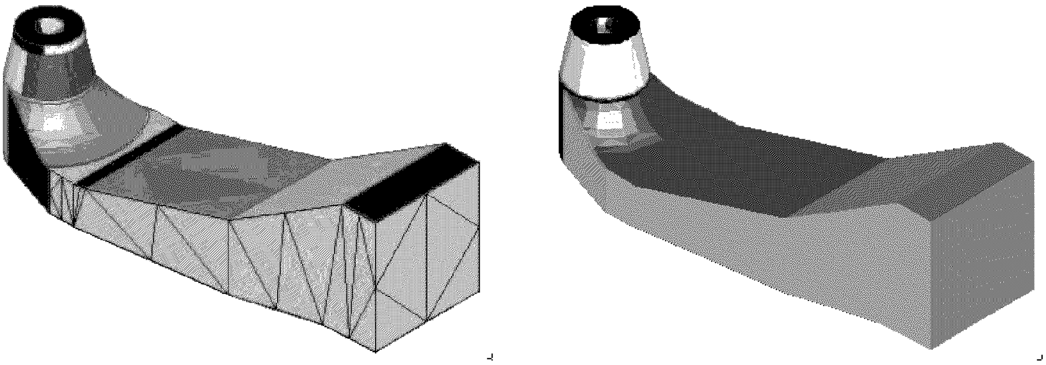


**Figure 2 Triangulated Parasolid™ CAD model (left), simplified model (right)**

# Automatic Hexahedral Mesh Generator

## Geometry adaptation

The HEXPRESS™ mesh generator is based on a volume to surface mesh approach. The methodology is described in details in Delanaye *et al* (2000), a short description is presented in the following. An initial mesh surrounding the computational domain is created. This mesh is not conforming to the geometry in most cases. The cell sizes of this initial mesh are most of the time not compatible with the local length scales of the geometry. Mesh adaptation is performed by successive subdivisions of cells in order to achieve clustering of points compatible with geometry length scales typical of high curvature regions, corners, ridges, etc. Further refinements and adaptations of the mesh are subsequently performed during the simulation depending on some indicators measuring the quality of the computed solution.

The local cell subdivision may result in the occurrence of neighboring cells with possibly very different sizes. Since those variations are incompatible with the accuracy of the numerical scheme, the difference of cell refinement levels across a common face is limited to a single level. This criterion advantageously forces the transport of refinement tags to neighboring cells and guarantees some level of smoothness in the mesh. In addition, the propagation of refinement tags from tagged cells to their neighbors and further in the mesh is controlled by a user defined diffusion depth parameter. Anisotropic subdivisions are moreover employed to avoid excessive growth of the number of cells as presented in Figure 3.

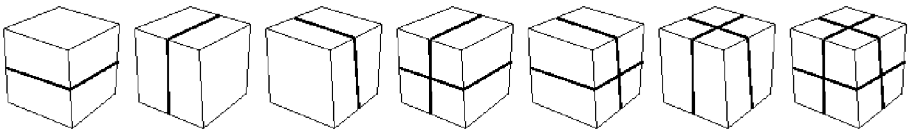


**Figure 3 Anisotropic subdivision of hexahedron**

At each adaptation iteration, the cells intersecting the geometry are searched. The cell sizes of the latter are compared to target cell sizes, which are defined by the user or automatically by the grid generator itself. Cells are refined if the criteria are not matched.

## Geometry Fitting

Once the non body-fitted grid has been sufficiently refined to match the typical length scales of the geometry, we proceed to the recovery of the geometry surface. Cells lying outside the computational domain or intersecting the surface geometry are marked for removal. Starting from a seed cell, a painting algorithm marks all the cells located outside of the computational domain; the latter are removed from the mesh. At this level, the boundary of the volume grid is a staircase surface inappropriate for flow simulations. Hanging nodes which can be present on this boundary due to the cell subdivisions are not removed since finite volume solvers are capable of handling such configurations.

Next, the staircase boundary of the volume mesh is snapped on the geometry surface. A Laplacian-like smoothing procedure is first applied to smooth out the staircase boundary. This smoothed boundary forms a front of quadrilateral facets whose vertices are projected on the geometry by a closest distance criterion. A closest projection point is accepted if the geometry normal computed at that location does not differ too much from the front normal at the corresponding facet vertex.

Important geometry features such as corners and ridges (*e.g.* trailing edges of wings) are not preserved by the snapping procedure. They are actually never present except sometimes if extremely fine grids are employed. For the accuracy of physics simulation, it is important to recover those special features. This step is crucial to produce a final mesh of high quality. Failure to choose the most appropriate vertices to attach to corners and ridges may create a mesh with distorted or even negative cells. The difficulty of associating a vertex to a corner or a curve is to make a choice in a set of several candidates that will eventually lead to the highest quality mesh without actually being able to measure the quality of the final mesh. The reader is referred to Delanaye *et al* (2000) for more details on the procedure. The projection of the smoothed staircase volume mesh boundary and the edge capturing produce angles close to 180 degrees for specific configurations. A layer of cells is extruded off the geometry surfaces and curves to remove these degenerate cells. The procedure is a generalization of the method of Mitchell and Tautges (1995).

## Optimization

Some degenerate cells may remain in the mesh after automatic mesh generation. These are due to the high distortion created in the mesh during geometry projection. The presence of those cells may hinder the convergence of the simulation tool or create negative cells during *h*-adaptation. Therefore, a very innovative optimization technique has been developed in HEXPRESS$^{\text{TM}}$. It consists in the successive applications of an algorithm which locally untangles concave cells and transform them in convex ones. The untangling algorithm applies to the set of triangles or tetrahedra which decomposes a quadrilateral or hexahedral cell respectively. An additional optimization algorithm improves the orthogonality of the cells by locally optimizing a functional defined on the convex cells. The reader is referred to Kovalev *et al* (2002) for more details.

## High aspect ratio cell layers

High aspect ratio cell layers are subsequently introduced to correctly resolve high shear flow phenomena such as boundary layers. This step can be performed either before any flow computation by the grid generator or during flow computation while adapting the mesh based on flow feature detectors. The user can specify several parameters such as first cell thickness, stretching value and number of layers. They are inserted by successively subdividing the buffer cells closest to the wall. The newly inserted vertices are then redistributed to match the specified distribution requirements. This creates a set of layers which are related to each other through the tree structure produced by the recursive subdivisions. This aspect is important for the performance of the multigrid solver because it allows the easy recovery of coarser cells through parent-child relationship.

# Adaptive Flow Solver

## Space discretization scheme

The spatial discretization method is based on a cell centered finite volume approach. The advective fluxes across a face are computed by flux averaging with added artificial dissipation (Jameson, 1995). The latter results in a blend of fourth and second order dissipation terms. A pressure switch triggers the second order dissipation factor in discontinuities or in very high gradient flow regions to avoid large amplitude oscillations. The calculation of the artificial dissipation term requires the computation of the solution first differences on the faces and of the second differences in cell centers (Van de velde *et al*, 1998). In a first loop over the cell faces, the variation of the solution vector across each cell face is calculated and stored. In a second loop, these variations across faces are transferred to the cell centers, with a plus sign for the upwind cell center and a minus sign for the downwind cell center. The viscous fluxes require the computation of temperature and velocity gradients on the cell faces. For this purpose, a diamond control volume is created around each face and consists of two pyramid elements. Each of them is formed by the face itself, and the left or right cell center as opposite summit respectively. For this purpose, the solution at the vertices is interpolated from the values stored at the cell.

The two equation $k$-$\varepsilon$ turbulence model (Jones *et al*, 1973) is used to simulate the effect of turbulence on the mean flow. The linear low-Reynolds model implemented in the code is due to Yang and Shih (1992, 1993). The particularity of this model resides in a redefinition of the turbulent time scale which removes the singularity at the solid wall. A wall function variant (Hakimi, 1997) of this model is also available. It dramatically reduces the number of mesh points required to resolve the boundary layer. The first mesh point should reside in the log-law region. The turbulent kinetic energy $k$ and its dissipation rate $\varepsilon$ in the cells next to the solid walls are updated according to formula based on DNS data, instead of solving the governing equations. In turbulent calculations, at inlet and outlet boundaries, the turbulent kinetic energy and dissipation rate are extrapolated from interior cells or imposed on the boundaries. On solid walls, the kinetic energy is zero, while the dissipation rate is again extrapolated from interior cells. In addition to the k-ε models, the one-equation turbulence model from Spalart and Allmaras (1992) has also been implemented.

## Multigrid acceleration

An explicit Runge-Kutta scheme integrates the discretized set of equations in time to eventually reach the steady state. Convergence acceleration is obtained thanks to local time stepping and multigrid acceleration.

In our multigrid approach, the creation of coarse grid levels is tightly coupled with mesh geometry and flow adaptation. Indeed, the initial mesh is used as the coarsest level, additional levels are created at each mesh adaptation either during geometry adaptation and/or flow adaptation by using the parent-child connectivity stored in the adaptation module. Since each multigrid level covers the whole computational domain, the final composite grid contains cells at many different refinement levels. The coarse multigrid level generation strategy used in the unstructured solver is derived from Aftosmis *et al* (2000), and consists in replacing all the leaf sibling cells by their parent. If one or more of a set of siblings has children of its own, then coarsening is suspended until those children are removed. In addition, in order to ensure sufficient mesh quality at each grid level, only one hanging node per edge is accepted. Therefore, a balancing function locally suspends the coarsening if a forbidden situation is found on the coarser level.

The grid transfer operators also use the cell parent-child connectivity. The restriction of the residual is simply chosen as the sum of the residual of the children, whereas a weighted averaging is used to restrict the solution. The prolongation operator interpolates the solution correction from a coarse level to a fine level. Basically, the correction is the difference between the new solution on the considered grid level and the restricted solution on the same grid level. A first order prolongation operator is used (Léonard *et al*, 1999). At first, the corrections in the cells of the coarse grid are interpolated in the vertices of the fine grid. In a second step, the corrections in these fine grid vertices are distributed to the cell centers.

## Flow adaptation

In the unstructured adaptive solver, mesh adaptation is performed automatically. The basic structure of an adaptive solution procedure consists in:

- Calculation of the solution on the current grid
- Identification of the cells to be refined and the cells to be removed
- Refinement or removal of the flagged cells

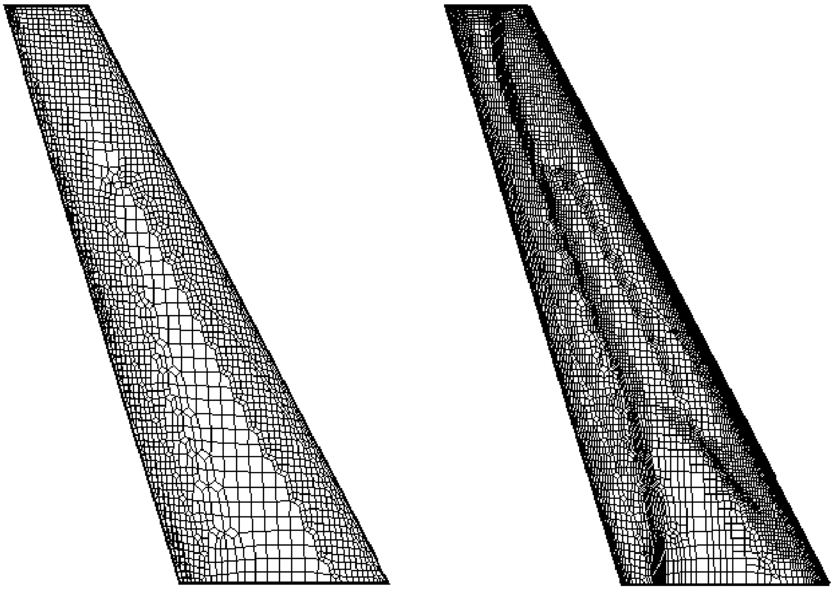
The anisotropic refinement functionality allows cells to be split in 2, 4 or 8. In order to ensure mesh quality, refinement flags are propagated to permit only one hanging node per edge. Furthermore, “islands and voids” in the mesh are prevented. A hierarchical mesh coarsening technique has also been integrated. To remove a cell, at least 75% of its siblings have to be flagged for coarsening. Then, the parent cell is recovered by removing all the siblings, including the non-flagged ones. As only one hanging node per edge is accepted, a balancing function is used to locally block the coarsening where forbidden configurations are foreseen.

Mesh adaptation is governed by criteria based on flow physics and geometry particularities. The first ones are flow feature sensors aiming at the detection of regions where significant flow variations exist. The choice of appropriate feature detection parameters is guided by the physical nature of the flow. Various criteria based on the flow physics are used. The undivided difference of pressure gradients is used to detect shock regions. Undivided and divided differences of the velocity magnitude as well as vorticity are used to capture viscous effects. No single sensor can adequately capture all flow features. An ideal sensor is usually defined for each testcase by combining several sensors. Refinement and coarsening threshold values are determined using a statistical formula (Kallinderis *et al*, 1989).

## Results

We first considered the simulation of inviscid and viscous flows around the LANN wing (Muller *et al*, 1996). In particular, the CT9 case is characterized by off-design conditions ($M_\infty = 0.82$, $Re_\infty = 7.17\ 10^6$, angle of attack equal to 2.6 deg.). This test case presents a strong interaction with a separated flow behind the strong shock system. Experimental data (AGARD AR-702, 1982) are provided.

An inviscid computation is carried out starting on an initial all-hexahedra unstructured grid involving 46435 cells (see Figure 4). The geometrical difficulty of this case is the presence of the very thin but blunt trailing edge, which needs to be resolved by the mesh. The mesh is adapted twice using finite differences of the velocity norm as adaptation criterion. After one adaptation, the mesh contains 128819 cells, and after two adaptations, it contains 211119 cells. After each adaptation, the number of grid levels used in the multigrid strategy is increased with a maximum of 3 levels.



Initial mesh, 46435 cells

Adaptation 2: 211189 cells

**Figure 4 LANN wing, adapted meshes**

Figure 5 presents pressure isolines on the surface. The solution actually better and better match the experiments (not shown) after each adaptation in the leading edge area, while the shock position is moved further downstream and the shock becomes crispier. Furthermore, the λ-shock structure becomes wider after each adaptation, *i.e.* the shock junction is moved further in the spanwise direction

The initial mesh is refined close to the wall in order to generate high-aspect ratio cell layers to resolve the boundary layer, 7 layers are added, the total number of cells now reaching 127675 cells. After carrying out one adaptation, the adapted mesh contains 233923 cells. Turbulence is initialized by assuming an initial value of 1% for the turbulent intensity. The pressure distributions computed on the fine mesh match better the experimental data than those computed on the coarser mesh, as shown in Figure 6 for the section located at 20 % and 32.5 % of the span.

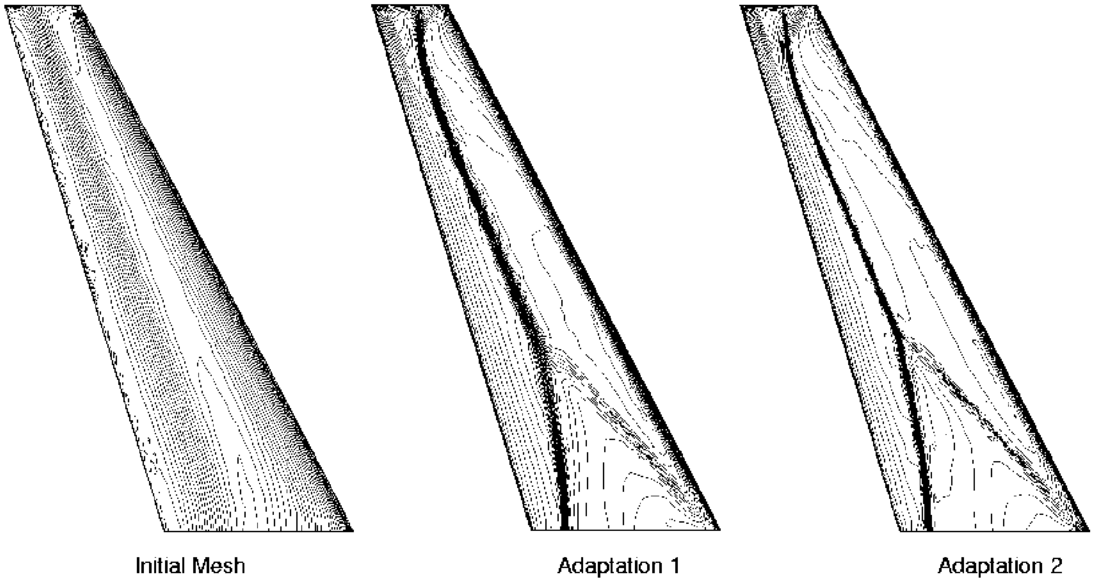


**Figure 5 LANN wing, adapted mesh solutions, pressure isolines**

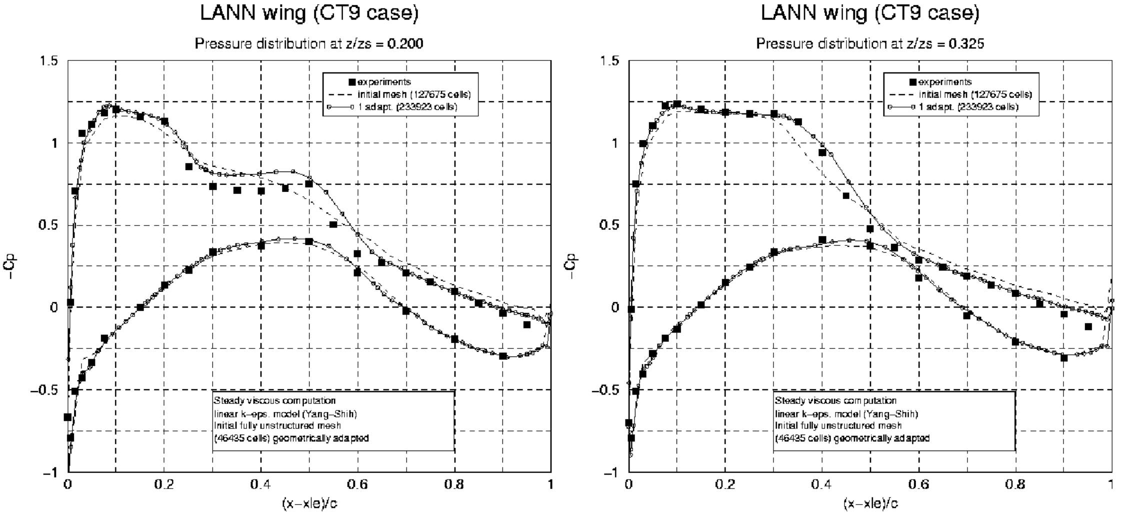


**Figure 6 LANN wing, surface pressure distributions, viscous flow simulation**

The second case is an aircraft wing-fuselage system referred to as the DLR-F4. This geometry has been recently analyzed in the framework of a CFD drag prediction workshop organized by the American Institute of Aeronautics and Astronautics. The flow is simulated at a Mach number equal to 0.75 and angle of attack equal to 0.93 deg. The unstructured hexahedral mesh contains 1402841 cells, the first cell size is 2.5 $10^{-5}$ m and a streching ratio of 1.2 is applied to the 15 layers of high aspect ratio cells used for the boundary layer resolution. The Spalart-Allmaras model is used to simulate turbulence. Figure 7 and Figure 8 represent the mesh and pressure isolines on the aircraft respectively. A converged solution is obtained in 500 cycles, using a full-multigrid approach to initialize the solution (Figure 9). Figure 10 presents the pressure distribution across two sections located at 0.238 and 0.844 fraction of the wing span respectively. They agree well with the results obtained with other codes in the context of the AIAA CFD Drag Prediction workshop.

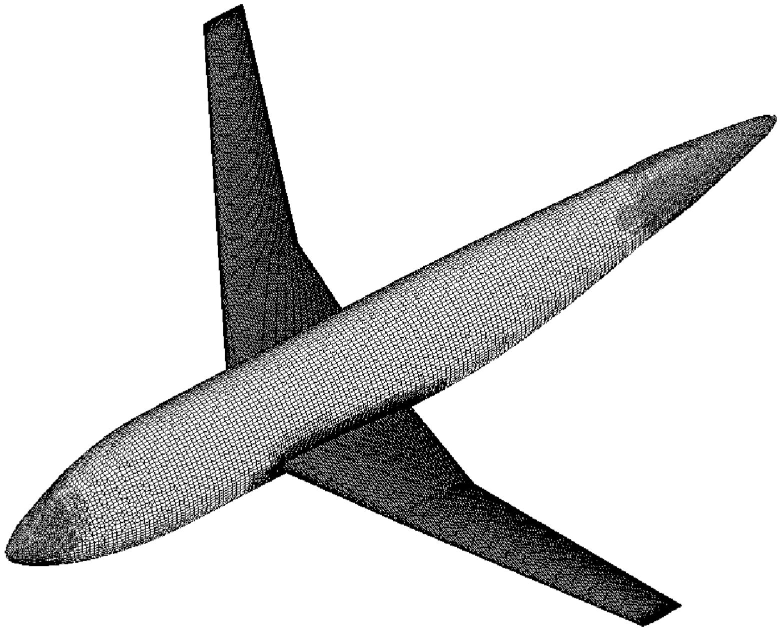


Figure 7 DLR-F4, 1402841 cells

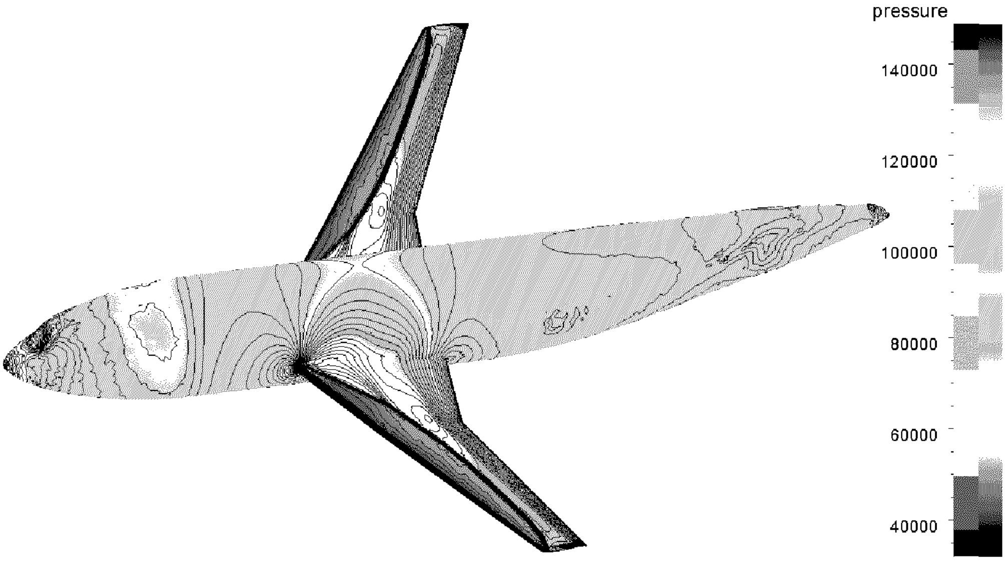


Figure 8 DLR-F4, pressure isolines

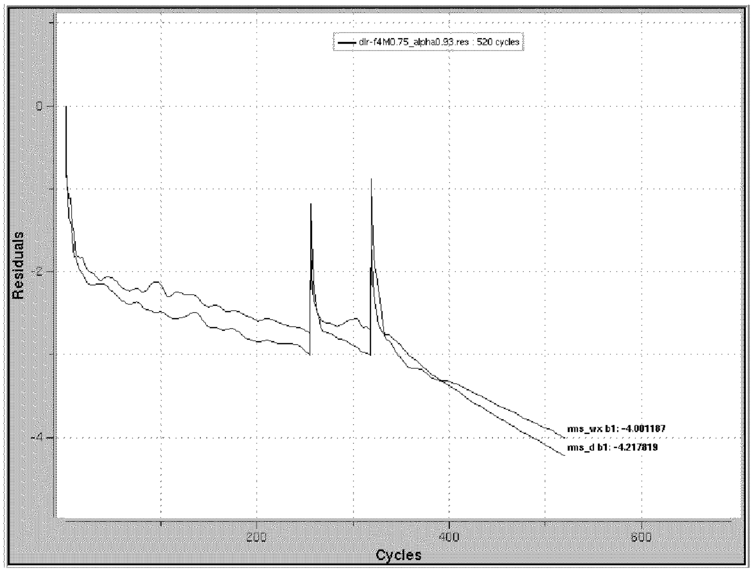


**Figure 9 DLR-F4, convergence history**

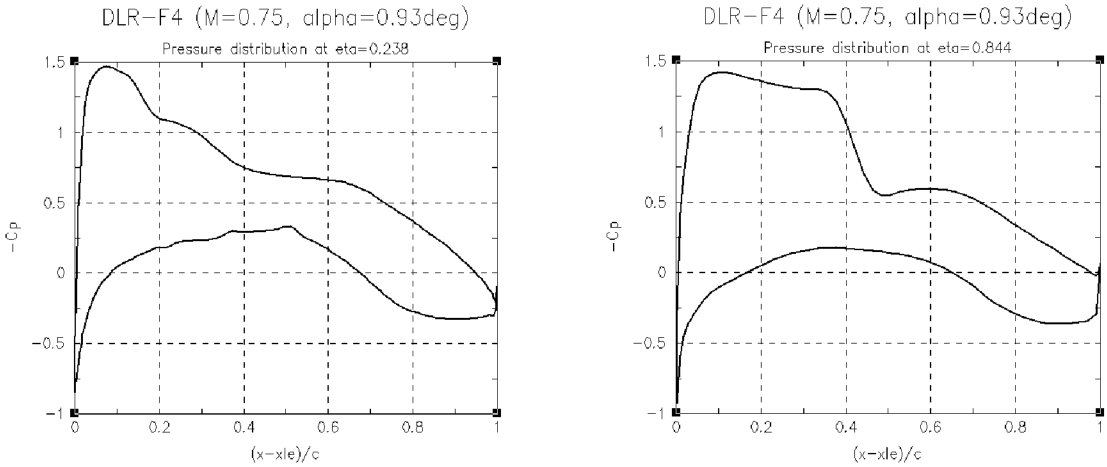


**Figure 10 DLR-F4: pressure distribution at two span sections**

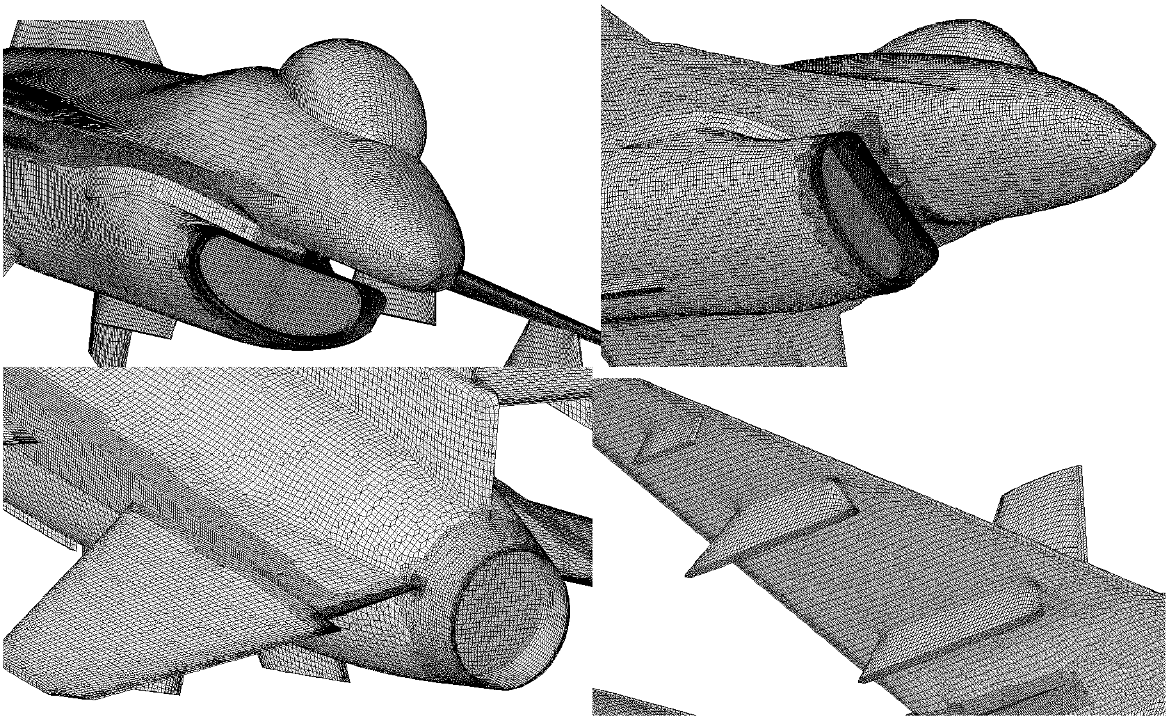


**Figure 11 F16 military aircraft. Unstructured hexahedral mesh, 432759 cells (half body)**

Figure 11 shows the mesh generated around a F16 military aircraft configuration. An inviscid flow simulation is carried out at a Mach number equal to 2 and no incidence. The mesh involves 432759 cells. The solution presented in Figure 12 and Figure 13 shows strong shock systems on the wings and fuselage. A residual drop of 3 orders of magnitude is obtained in about 200 cycles (Figure 14).

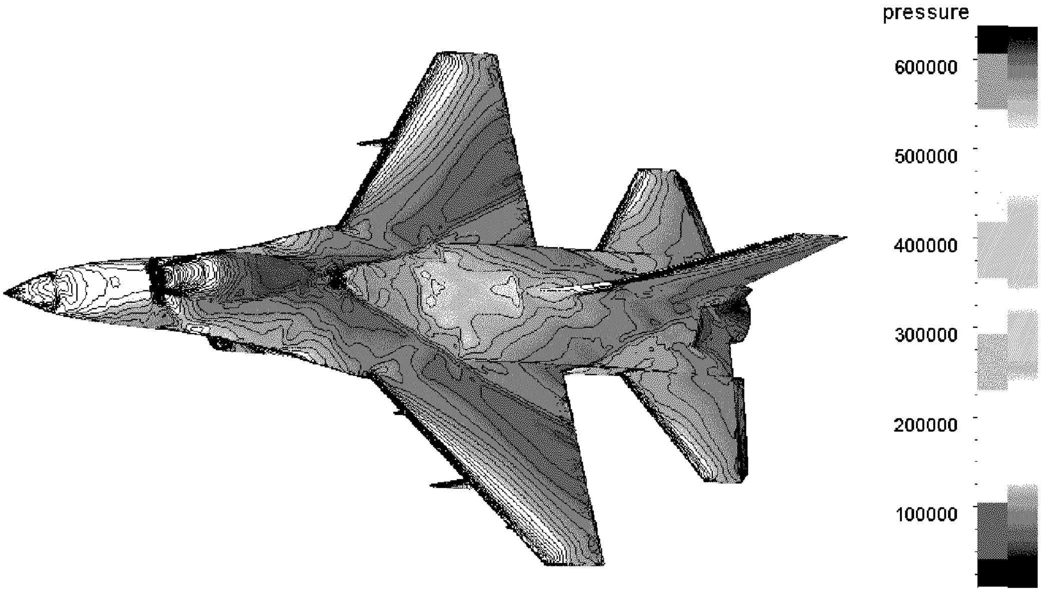


**Figure 12 F16 military aircraft at Mach number equal 2, angle of attack 0 deg. Pressure isolines.**

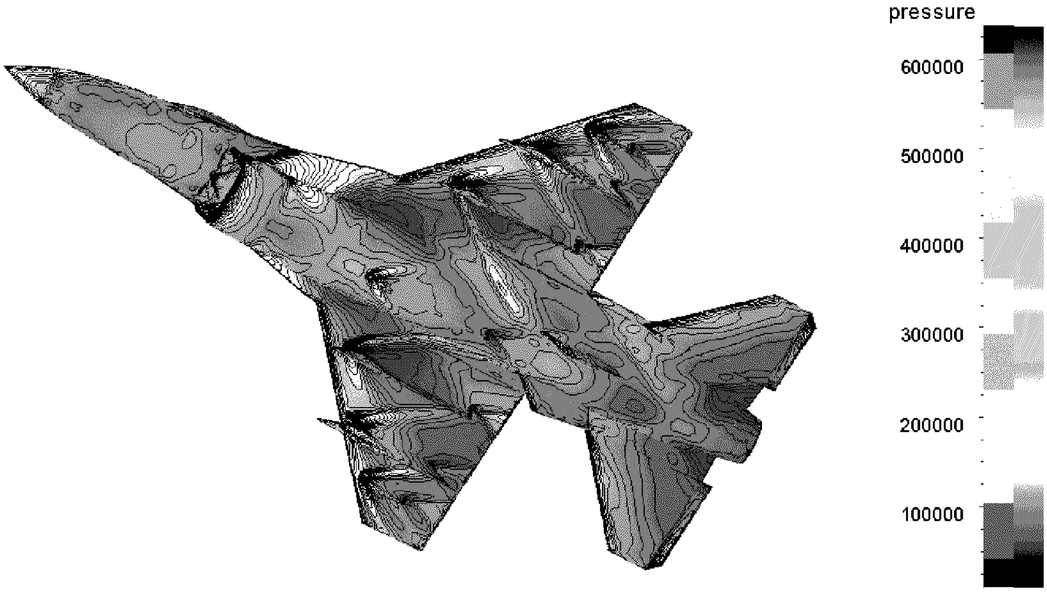


**Figure 13 F16 military aircraft at Mach number equal 2, angle of attack 0 deg. Pressure isolines.**

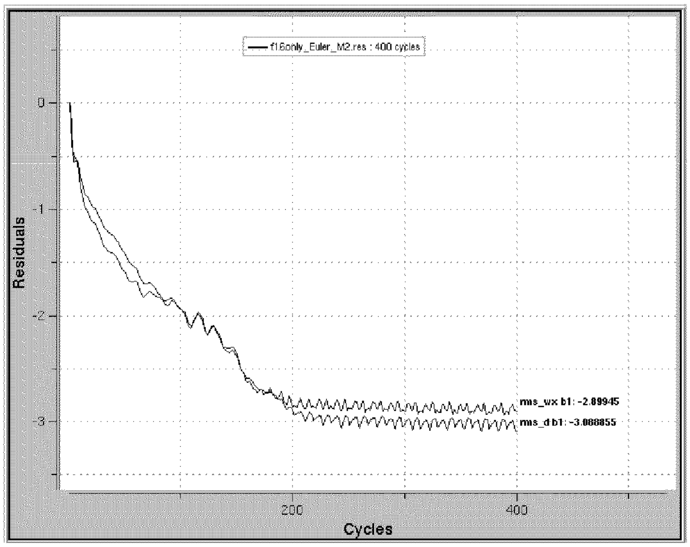


**Figure 14 F16 military aircraft at Mach number equal 2, angle of attack 0 deg. Convergence**

# Conclusion

An error controlled system for flow simulation around complex geometries is presented. The approach is based on automatic hexahedra meshes adaptive flow. Hexahedral meshes present the advantage of preserving the accuracy of well known numerical methods developed for structured meshes, they also minimize the number of cells used to resolve boundary layers for complex geometries. Hexahedra cells can be easily decomposed anisotropically which results in a powerful adaptation technique for flows presenting very different scales. The inherent tree structure resulting from mesh adaptation by successive subdivisions to geometry and further to the flow solution is exploited to devise a fast multigrid convergence acceleration method. These advantages have led to the development of a powerful environment, tightly integrated through a common object oriented language (C++) capable of solving very complex flows in complex geometries of industrial interest.

## Acknowledgements

The authors would like to thank the Region of Brussels Capital for its support through the R\&D projects 95-B-168 and RBC-BR 196/4155. The European Community is also acknowledged for its support through the Chamad Esprit Project 25059, Brite/Euram project BRPR-CT97-0583 UNSI and the ESTEC Contract No 12110/96/NL/FG.

## References

Schneiders, R., A Grid-Based Algorithm for the Generation of Hexahedral Element Meshes, *Engineering With Computers*, 12, 168-177, 1996.

Delanaye, M., Tchon, K.-F., Patel, A. and Hirsch, Ch., All Hexahedra Unstructured Grid Generation, *in Proc. of ECCOMAS 2000*, Barcelona, 2000.

Aftosmis, M.J., Solution Adaptive Cartesian Grid Methods for Aerodynamic Flows with Complex Geometries, *Computational Fluid Dynamics VKI Lectures Series 1997-05*, 1997.

Taghavi, R., Automatic, Parallel and Fault Tolerant Mesh generation from CAD, *Engineering with Computers*, 12, 178-185, 1996.

Mitchell, C.R. and Tautges, T. J., Pillowing doublets: refining a mesh to ensure that faces share at most one edge, *in Proceedings of the 4th International Meshing Roundtable*, 231-242, Albuquerque (NM), 1995.

Jameson, A, Analysis and design of numerical schemes for gas dynamics', *Int. J. Comp. Fluid Dyn.*, 4, No 3-4, 171-217, 1995.

Kovalev, K., Delanaye, M. and Hirsch, Ch., *Untangling and Optimization of Unstructured Hexahedral Meshes*, paper to be presented at the Grid Generation Theory and Applications Workshop, Russian Academy of Sciences, June, 2002.

Van de Velde, O., Lacor, C., Alavilli, P. and Hirsch, Ch, A 3D unstructured Navier-Stokes solver with multigrid on adaptive hexahedral meshes, *in Proc. 4th European CFD Conf.*, 675-680, 1998.

Jones, W.P. and Launders, B.E., The calculation of low-Reynolds number phenomena with a two-equation model of turbulence, *Int. J. of Heat and Mass Transfer*, 106, 119 (1973)

Yang, Z. and Shih, T.H., A k-ε calculation of transitional boundary layers, *NASA Technical Memorandum 105604 ICOMP-92-08*, CMOTT-92-05, 1992.

Yang, Z. and Shih, T.H., A Galilean and tensorial invariant k-ε model for near wall turbulence", *AIAA Paper 93-3105*, 1993.

Spalart, P. and Allamaras, S.R., A One-Equation Turbulence Model for Aerodynamic Flows, *AIAA paper 92-0439*, 1992.

Hakimi, N, Preconditioning methods for time dependent Navier-Stokes equations / Application to environmental and low speed flows, *PhD, Vrije Universiteit Brussel,* 1997.

Aftosmis, M.J., Berger, M.J. and Adomavicius, G., A parallel multilevel method for adaptively refined Cartesian grids with embedded boundaries, *AIAA Paper 2000-0808*, 2000.

Léonard, B., Patel, A. and Hirsch, C., Multigrid acceleration in a 3D Navier-Stokes solver using unstructured hexahedral meshes, *in Proc. 4th Conference on Multigrid*, 1999.

Kallinderis, Y. and Baron, J.R., Adaptation methods for a new Navier-Stokes algorithm, *AIAA Journa*l, 27, No 1, 37-43, 1989.

Muller, U.R., Henke, H. and Schulze, B., Computation of Transonic Steady and Unsteady Airfoil and Wing Flow by Inviscid and Viscous-Coupled Solvers", European Computational Aerodynamics Research Project: Validation of CFD Codes and Turbulence Models, Notes on Numerical Fluid Mechanics, *Eds Haase, Chaput, Elshoz, Leschziner, Muller, Vieweg*, 58, pp. 81-88, 1996.

AGARD, Compendium of Unsteady Aerodynamic Measurements, Addendum No.1, *AGARD AR 702*, 1982.

UNSI Brite Euram European Project - Final Report, *Vieweg*, (to be published).